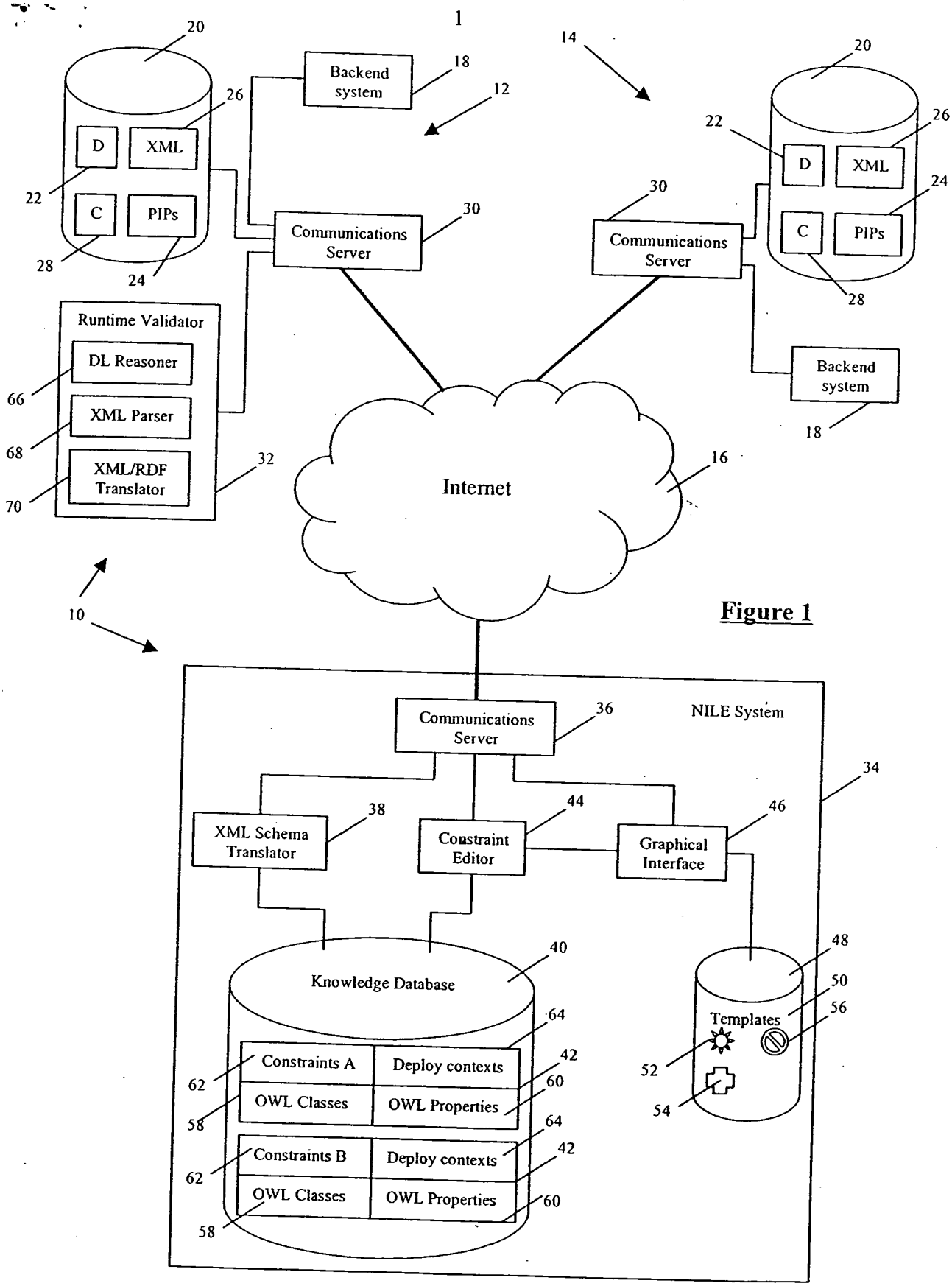
1
20
Backend system
18
12
14
20
26
22
26
D
XML
22
D
XML
24
C
PIPs
30
Communications Server
30
Communications Server
C
PIPs
28
24
28
Runtime Validator
DL Reasoner
66
XML Parser
68
XML/RDF Translator
70
32
Internet
16
Backend system
18
10
Communications Server
36
NILE System
34
XML Schema Translator
38
Constraint Editor
44
Graphical Interface
46
Knowledge Database
40
48
50
56
Templates
52
54
64
Constraints A
Deploy contexts
42
62
60
OWL Classes
OWL Properties
58
64
Constraints B
Deploy contexts
42
62
60
OWL Classes
OWL Properties
58

Figure 1

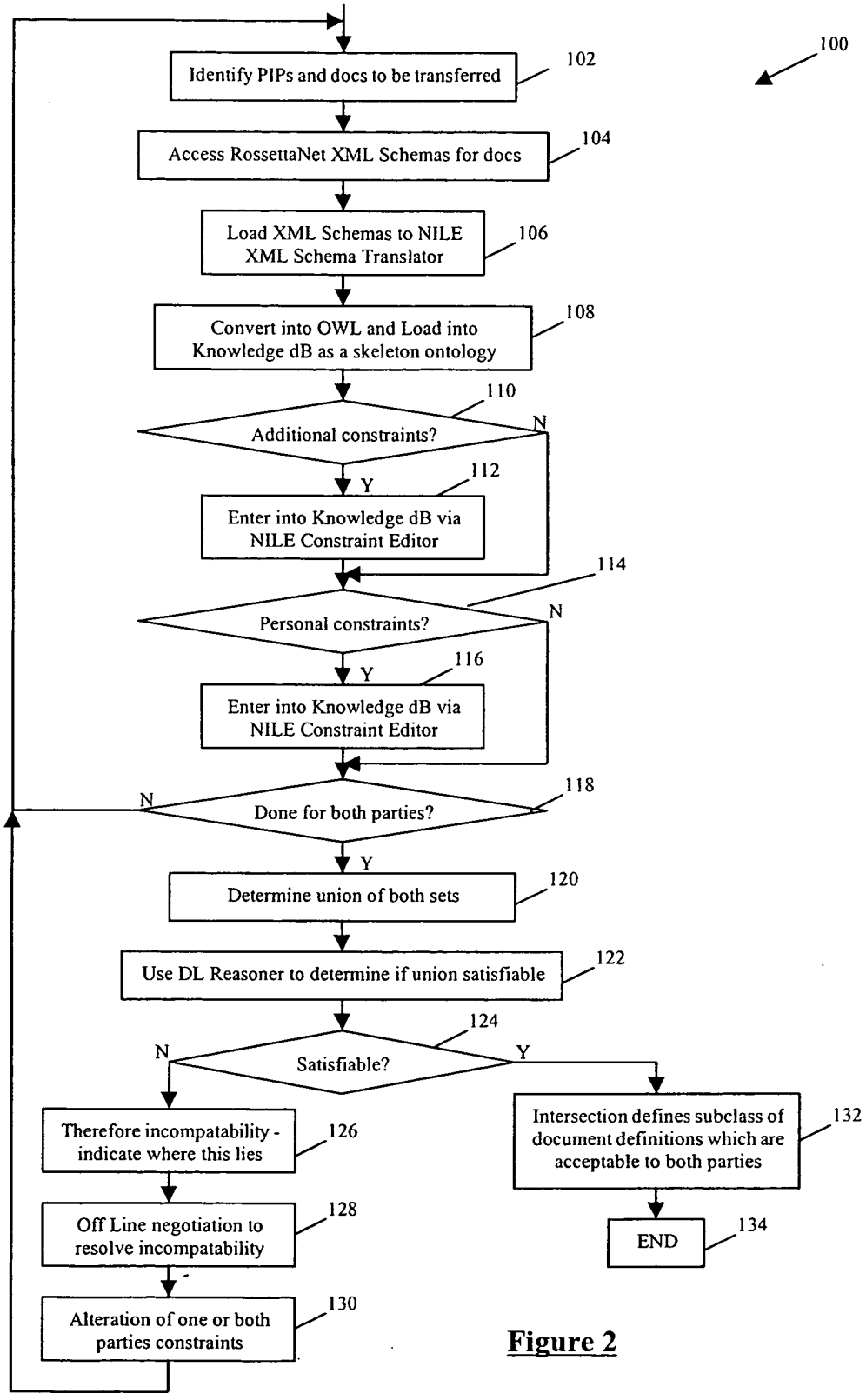
100
Identify PIPs and docs to be transferred
102
Access RossettaNet XML Schemas for docs
104
Load XML Schemas to NILE XML Schema Translator
106
Convert into OWL and Load into Knowledge dB as a skeleton ontology
108
Additional constraints?
110
N
Y
Enter into Knowledge dB via NILE Constraint Editor
112
Personal constraints?
114
N
Y
Enter into Knowledge dB via NILE Constraint Editor
116
Done for both parties?
118
N
Y
Determine union of both sets
120
Use DL Reasoner to determine if union satisfiable
122
Satisfiable?
124
N
Y
Therefore incompatability - indicate where this lies
126
Off Line negotiation to resolve incompatability
128
Alteration of one or both parties constraints
130
Intersection defines subclass of document definitions which are acceptable to both parties
132
END
134

Figure 2

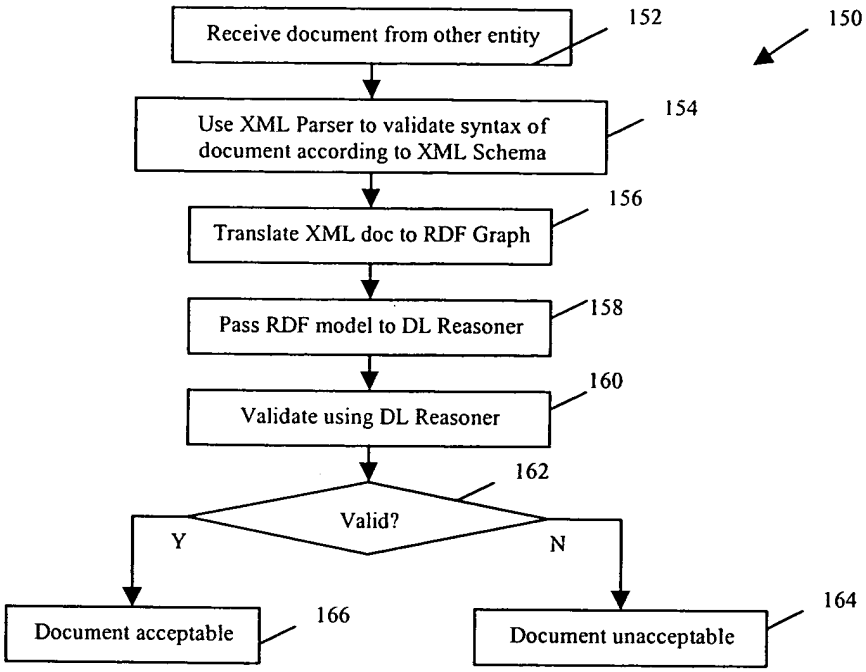

Figure 3

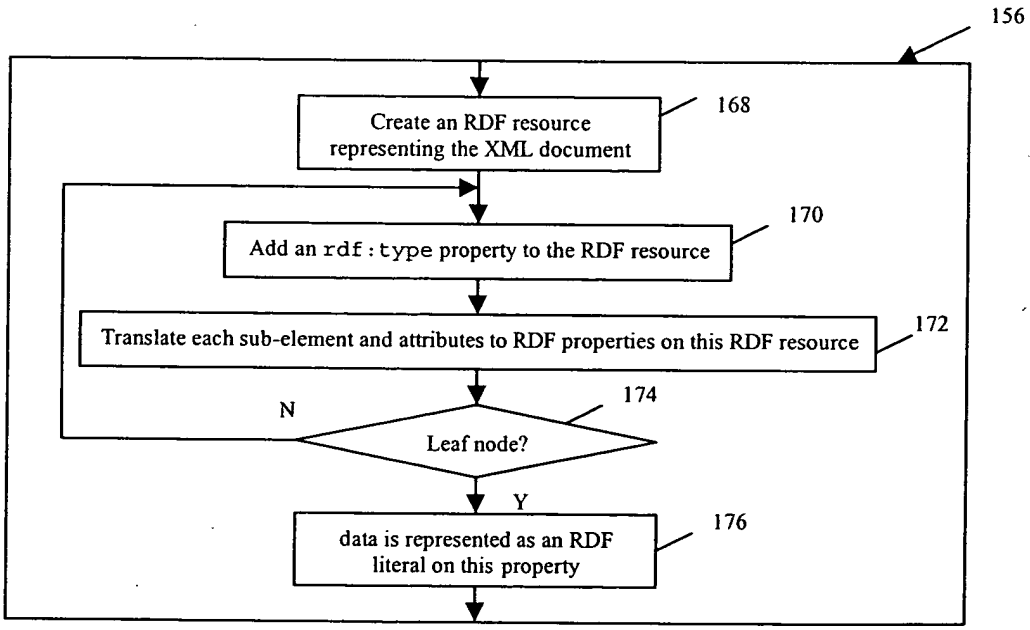

Figure 3a

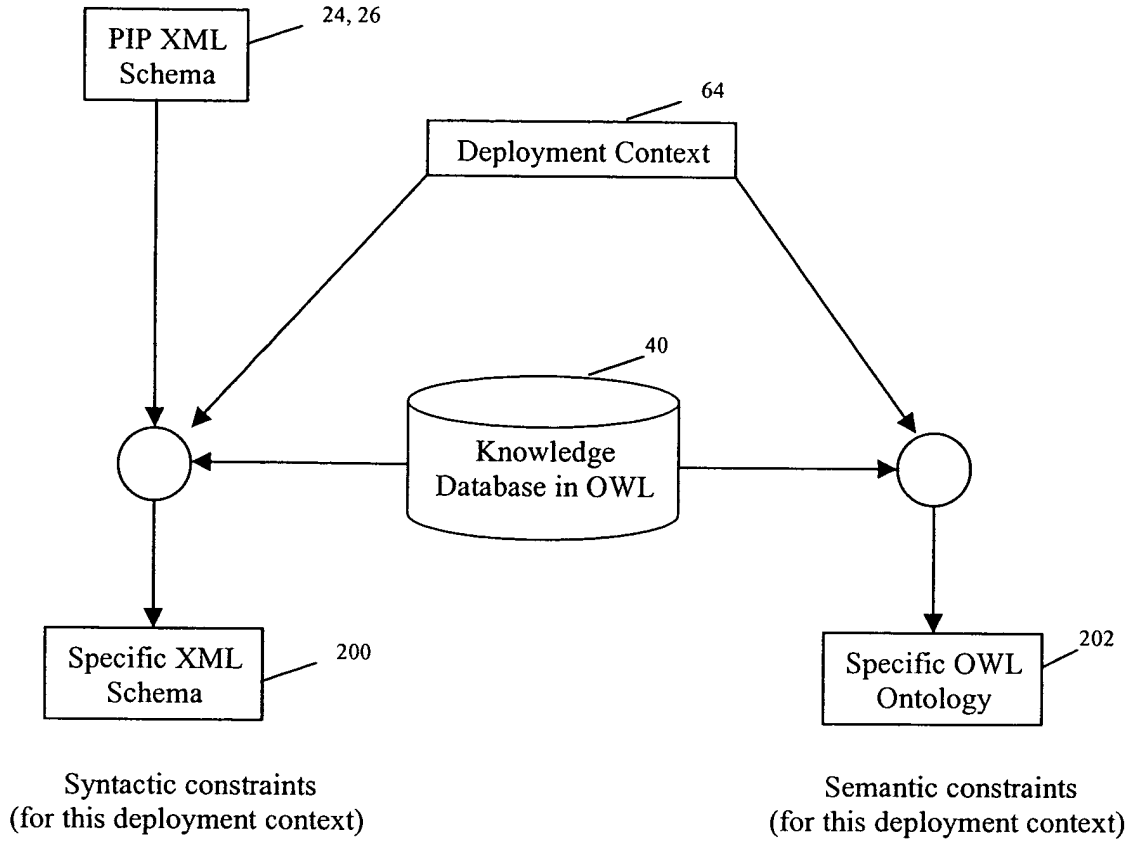

Figure 4

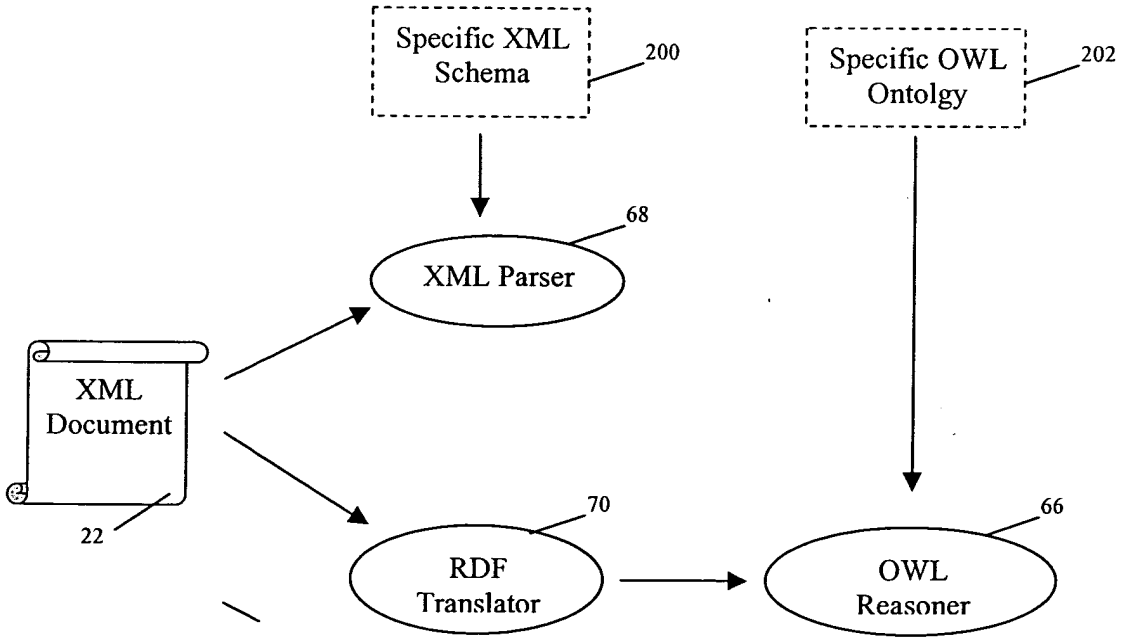

Figure 5

1. The root `schema` element, `complexType` definitions, model `group` definitions and `attributeGroup` definitions are mapped to OWL classes.

2. Named `simpleType` definitions stay untouched; anonymous `simpleType` definitions are assigned a unique name and copied to a separate datatype file. These `simpleType` definitions are used to restrict datatype properties.

3. `complexType` elements are mapped to OWL object properties; `simpleType` elements and attributes are mapped to OWL datatype properties.

4. Type and occurrence specifiers of elements and attributes are mapped to an intersection of OWL property type (i.e. `toClass`) restrictions and cardinality restrictions.

5. `extension` and `restriction` definitions are mapped to an OWL `subClassOf` relationship.

6. Groups with a `choice` compositor are mapped to the OWL equivalent of an XOR (with `intersectionOf`, `unionOf` and `complementOf`).

7. Groups with an `all` or `sequence` compositor are mapped to an OWL `intersectionOf` collection.

8. `substitutionGroup` relationships are mapped to an OWL `subPropertyOf` relationship.

9. Names of components are always mapped to an URI composed of the schema `targetNamespace`, # and the component's name.

Figure 6

Constraint	Deployment Context			
	Document	Sender	Receiver	Backend System
A	PIPC3	NewCo	A	SAP1
B	Any	A or B	C	All SAP
C	PIPC2/3 or PIPC7	NewCo or A or C	B or D	All SAP or all Oracle
D	EPPartners and PIPC4	B	A or C	SAP1 or SAP2
E	USClients	D or C	NewCo or A	SAP1
...	...	...	...	...
...	...	...	...	...

62, 64, 300, 302, 304, 306, 308, 308, 308

Figure 7

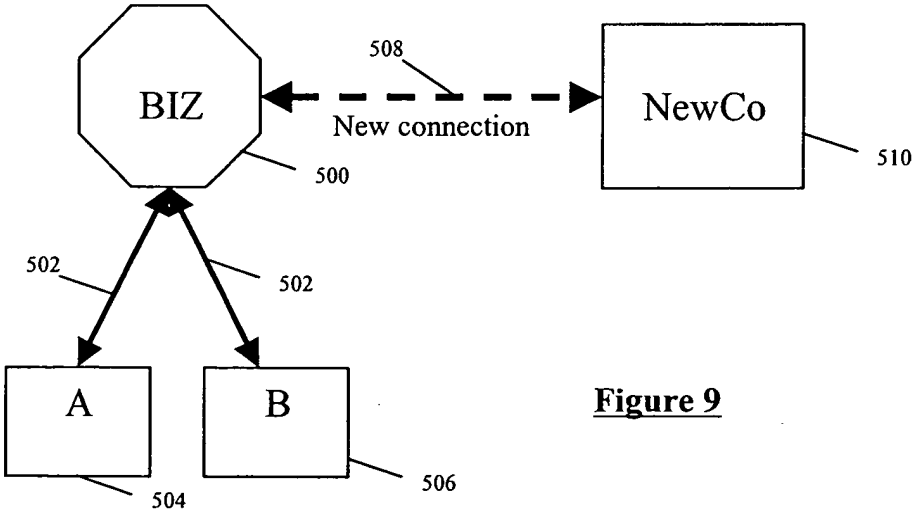

Figure 9

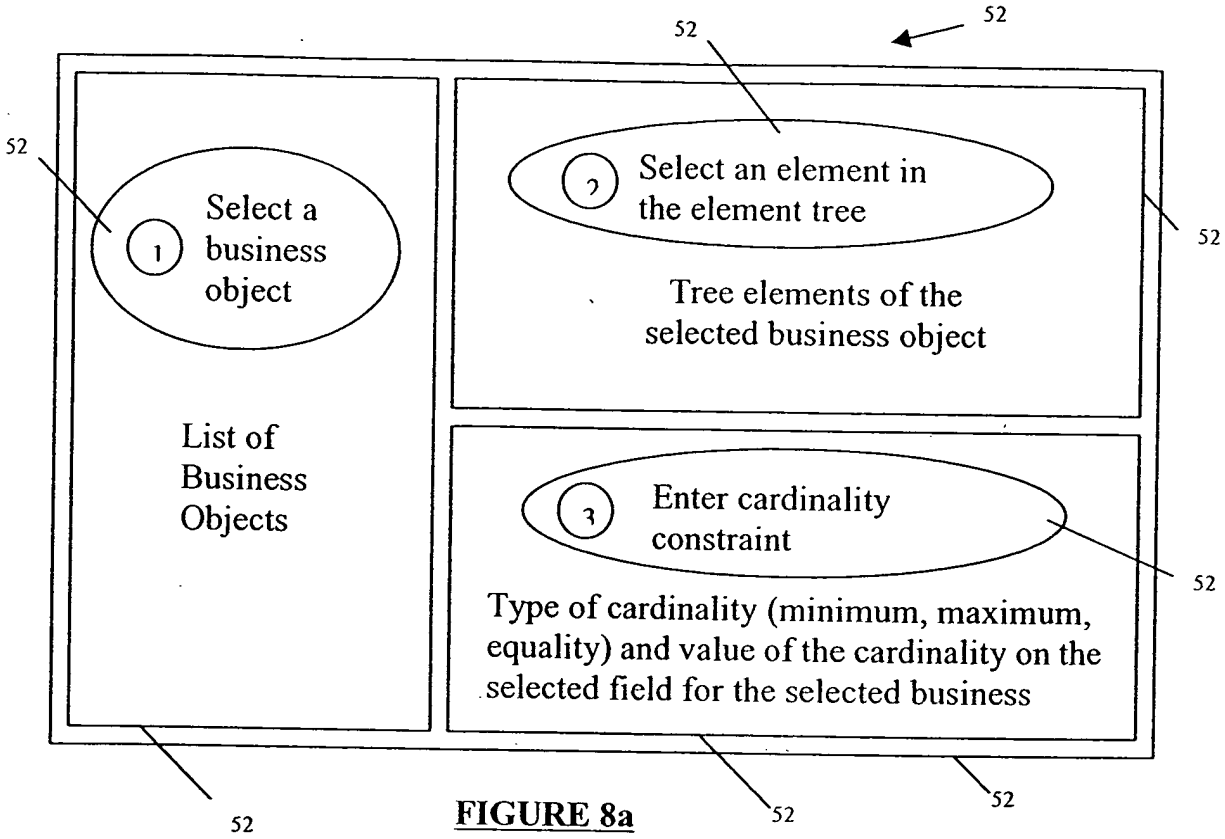

FIGURE 8a

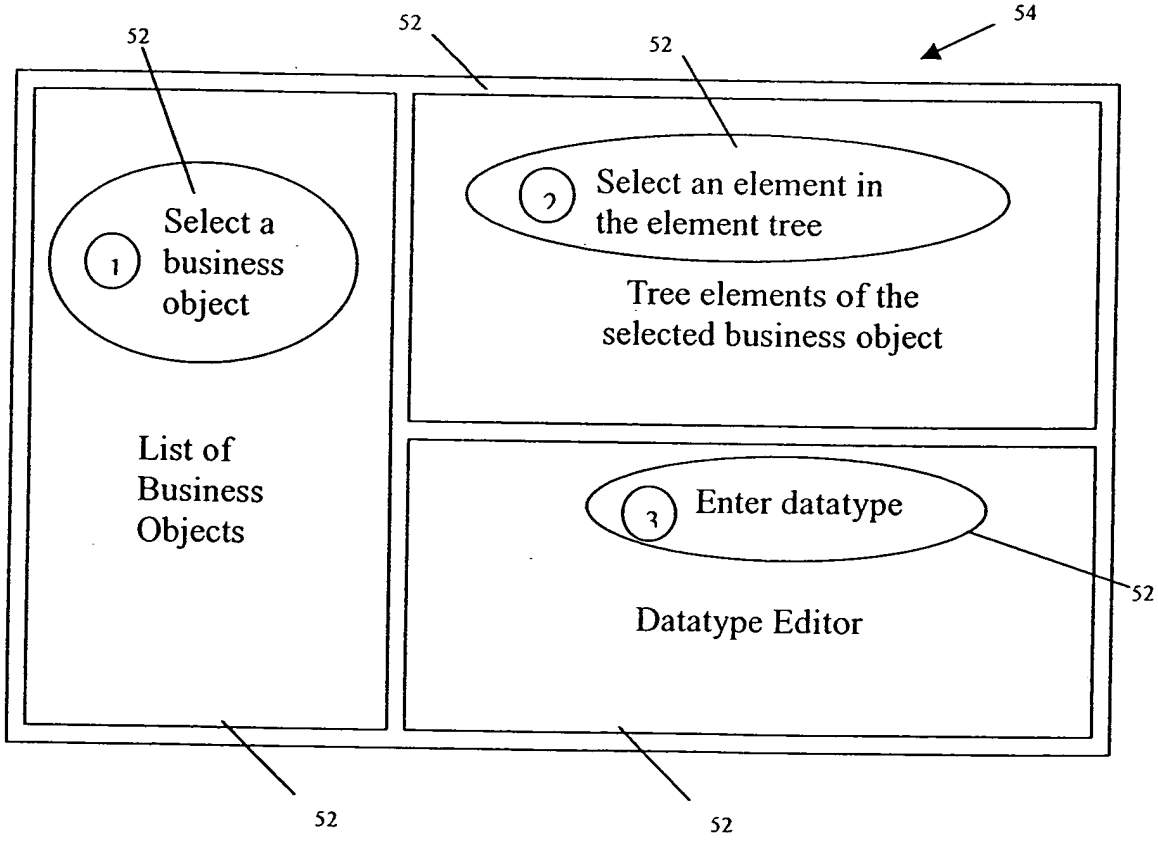

FIGURE 8b

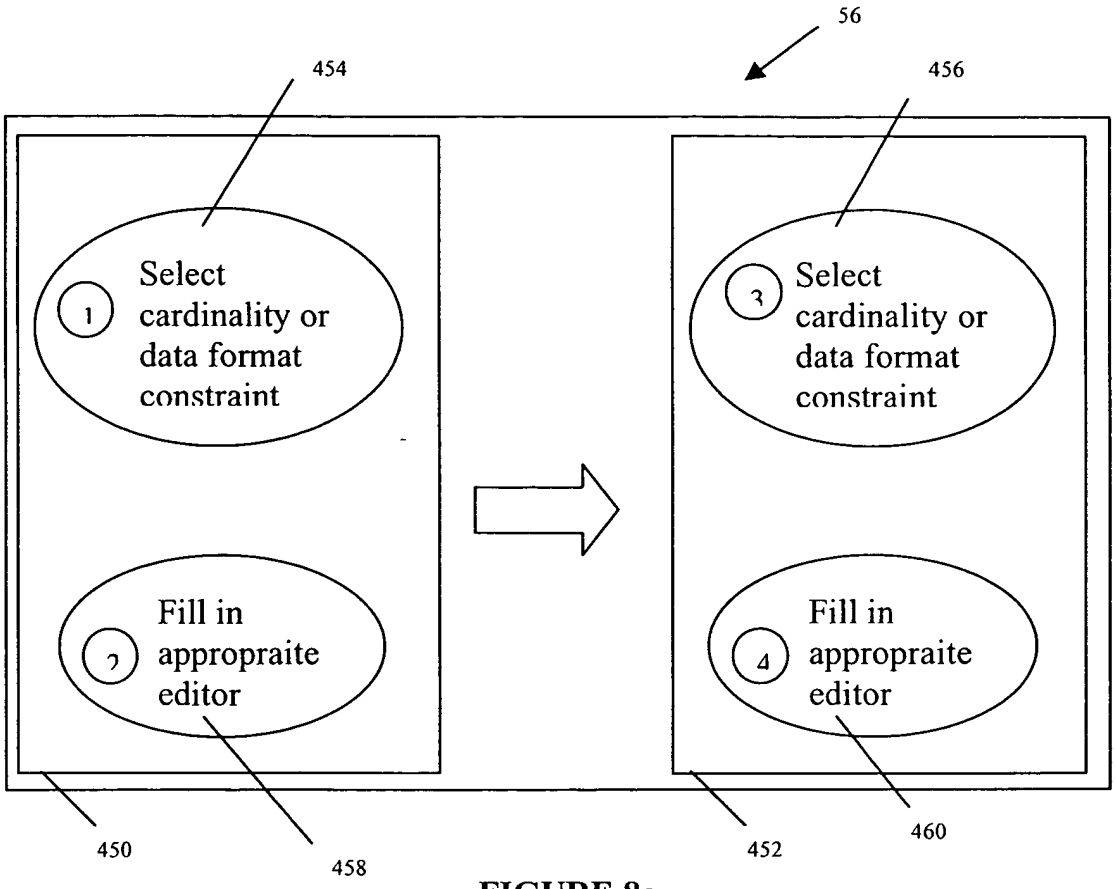
56
454
456
1
Select cardinality or data format constraint
2
Fill in appropraite editor
3
Select cardinality or data format constraint
4
Fill in appropraite editor
450
458
452
460

FIGURE 8c